VARIABLE NAMES AND ADDRESSES WITH THE AERCO 2068 DISC INTERFACE

W. L. Fadner
Department of Physics
University of Northern Colorado
Greeley, CO 80631

As mentioned in the evaluation of the Aerco 2068 Disc Interface in the Sept.-Oct. issue of SyncWare News, the Aerco system does not presently support commands which have variable names or addresses when loading or saving disc files. These include commands such as CAT "a$.bin", or MOVE "a$.bin",y,L where the variables for the name, a$, the beginning address, y, and the program length, L, have all previously been specified in the program. Almost any program in which the user generates files needs to make use of commands of this type. That includes word processors, data bases, spread sheets, graphics programs, and others. Aerco has said that they will attempt to incorporate that facility in future EPROMS and boot discs, but the date for this is indeterminate.

In the meantime this shortcoming must be bypassed through programming. Unfortunately, this is a somewhat difficult task. A method for doing this is to put a MOVE or CAT command having a temporary name and locations in the program at the appropriate line location. Then when the program runs, it must change that BASIC line after the proper title and numbers have been determined, but before the line is executed. The program procedures given below for this process may be used by individuals in programs for their own use, but the author reserves all commercial rights. The following program, "Line Change Test Program" shows how the method works, and can give some needed information for a specific required line change.

(EDITOR: Insert "Line Change Test Program Here)
(INSERT B)

The actual line to be changed is number 9055. It is never executed in this test program, but is listed both before and after the change. Line 9054 obtains the address of line 9055 by peeking at memory addresses 23637 and 23638, which give "NXTLIN". Whatever poking is to be done to line 9055 must be done in line 9054, or in subroutines called from 9054. No other lines can intervene. In this test program, the printer writes out a number of the actual elements in line 9055, after asking you how many elements you want to see. This can be used as an aid in counting properly for a POKE statement to change the line. As presently constituted, the program writes on the 2040 printer, but this could be changed to the screen by changing all LPRINT commands to PRINT.

The values of the variables a and b would normally be obtained from the program. These must be broken into separate digits for poking, and this is done in subroutine 3000-3020, along with lines 2010 and 2030. Normally, the name to be poked would be typed in by the user. Here it is NewNamexxx, represented by the variable a$. Note that for this procedure to work, all titles must be of the same length, or spaces must be added to each title to ensure they all are the same length. Line 2050 (in a subroutine) pokes the name and numbers into line 9055. Line 1055 does nothing whatsoever except remind you of what line 9055 used to look like. If you LIST the program after running it, you will find that line 9055 is no longer what it was, like line 1055.

Following is the output from this program, after typing the number "35" in for the number of elements, or bytes, in the line. In the output, the first five elements are (as with any line) as follows.

(1) and (2): The number of the line in two-byte decimal, the first number being the most significant (9055= 256*35 + 95).

(3) and (4): the number of elements (bytes) in the line, including two elements for the line number, in two byte decimal with the least significant number first.

(5): always 32, code for a space.

This is the same format which is mentioned in the Timex 2068 Technical Manual, page 119.

(EDITOR: Insert Test Program Output here)
(INSERT C)

In future use of the above program, a person should put a STOP statement in line 9054, just after "GO SUB 1000". Then RUN the program to obtain just 4 line elements. The third and fourth elements in the line give the total number of elements in the line. Rerun the program for that total number of elements. Count out the element locations in order to construct line 2050 or its equivalent for proper poking of line 9055. Then delete the temporary STOP statement and test run to see if the poking proceded properly. Of course, the proxy line like number 9055 must always have the exact same spacing as the final disc command you need. Now you can put the equivalents of lines 9055, 9054, and 2050 into your main program.

An example of the use of this technique will be included in the next issue, when modifications for a disc version of Tasword II (the word processor) will be presented.

© 1985 W. L. Fadner

MODIFICATIONS TO TASWORD II TO WORK WITH THE AERCO DISC INTERFACE

W. L. Fadner
Department of Physics
University of Northern Colorado
Greeley, CO 80631

In the last issue, a method of using variable titles and address numbers with the Aerco Disc Interface was presented. As a practical application of this method, modifications to TASWORD II to work on the AERCO disc system are shown below. These result in a word processor which works extremely well. As was the case for the previous article, individuals may use these program segments for their own private use, but commercial rights are reserved.

MERGE The TASWORD II basic program from tape into memory, as described in the Sept.-Oct. SyncWare News article on the Aerco Disc System. Store this on a disc before modifying it. (I called this TasTape.BAS.) We will need two modifications of the program, for reasons to be given shortly. To obtain the main BASIC Tasword program, make the following changes.

1. Delete the individual lines 11 and 900.
2. Delete the following groups of lines

```
DELETE 200,490
DELETE 600,800
DELETE 1000,3060
```

3. Now type in the following lines, which involve changes to previous lines or new lines to be added. An alternative would be to first save the lines given below on tape as a program and then merge it into the program segment obtained from step 2, just above. Of course, the lines with "No Change" statements should not be typed in. (Note that there are 10 spaces in line 1022. This is part of a routine to make all titles a total of 10 spaces long, including blanks.)

(EDITOR: Insert Tasword Change Program #1 here)
(INSERT D)

4. This modified program can now be saved on tape with the command,

```
MOVE "a:Tas.bas",15
```

The above program handles all of the BASIC operations except printing (lines 200-290 in the original Tasword) and setting up the graphics commands and printer interface commands (lines 300-490 in the original Tasword). It was not possible to include these in the new main BASIC program within the space available. Ramtop could not be set significantly higher because the binary text file and binary operating program had to fit above ramtop.

The solution used here for this space problem (a solution which could be used in many other programs) was to put the two operations just mentioned into a separate BASIC program which is loaded only when the user selects these options using the menu. Then this secondary program, called "SubTas.BAS", is loaded, erasing the main "Tas.BAS" program. Since the binary programs are not affected by the change in the BASIC program, a person only needs to be concerned with any BASIC variables that must be carried over from one program to the next. In this case there were four such variables. In line 15, default values for d$ (the disc drive address for text files), p$ (lines/inch for the printer), and s$ (1 or 2 line spacing) are poked into locations 54786, 54787 and 54788. In line 20, the value of a, the length of the text program, is poked to memory locations 54784 and 54785. All of these addresses are previously unused locations at the beginning of the Tasword binary program. These values are used (or changed) in both the main and secondary BASIC programs.

When the operations in the secondary BASIC program are completed, the main program is reloaded from the disc (lines 295 and 490 in the secondary program), with starting line 18, inserted for that purpose. Loading either program takes about 1 second. That procedure would clearly be impractical or impossible using tape inputs and outputs. As a bonus, there was enough space in the secondary program to add several additional facilities to the printing operation.

The secondary program is obtained through the following process.

5. Load the original Tasword tape program into the computer. Then go to the menu and basic operation and execute the following:

```
DELETE 10,190
DELETE 500,800
DELETE 1000,3060
900            (delete line 900)
```

6. Type in the following lines. Alternatively, you could use the merge method discussed in step 3. Again, the lines stating "No Change" should not be typed in as shown.

(EDITOR: Insert Tasword Sub Program Changes here)
(Insert E)

7. Save this new program with the command

```
MOVE  "a:SubTas.bas",
```

Now it is only necessary to obtain the binary program and save it on the disc.

8. Load the binary program from the tape:

```
LOAD "" CODE
```

9. Execute MOVE "a:Tas.bin",54784,10751 The Tasword program should now be ready to use, with the command:

```
CAT "a:Tas.bas",
```

If you have not initialized the program for use with your printer, you should do that now. Also, you should set up any graphics commands. It is expected that you will use "Top of Form" (which I put on the solid square symbol). You might also use underlining, different type styles, subscripts and superscripts, etc. After making these adjustments, then save Tasword using the menu command. The new version of the program, with these adjustments, will be saved on the disc.

In every case, all Tasword programs are saved on disc A. User generated text files are stored and retreived from disc B, unless you choose a different disc drive. You will have to modify the program for the use of disc drives C and D, if you have these.

In the printing routine, the operator is given a choice of draft or letter quality printing. The codes used for that, in the LPRINT functions in lines 271 and 272 of SubTas.bas, are compatable with the Epson FX-80 and the modified Okidata ML-82A mentioned below. You would have to change these code sequences for some other printers.

The number of lines per inch (6 or 8) are chosen in the printer parameters routine, lines 500-598. This choice is executed in line 273 or 274 of SubTas.bas, where both the length of the page and the number of lines per inch are set. The codes in those two lines are compatable with the modified Okidata, and would have to be changed for some other printers. The line spacing: single(1) or double (2), or greater, is also set in the printer parameters routine. This is executed in line 257 and will work with any printer for which Tasword has been initialized. Whenever the text is saved, these parameters are poked into the addresses for the first two symbols of the text (in lines 1050 and 1060 in the main program). This routine was used because a person normally tailors the page length of the text according to the lines per inch and line spacing he intends to use. So, the parameters should be stored with the text. When the text is loaded back into Tasword, these codes are read, and the corresponding parameters are set into the computer (lines 2120-2135 in the main program). The codes are then erased from the text. The only disadvantage of this method is that any text you had put in the first two spaces of each file will be erased. If you merge a text into Tasword, these codes in the text will not make any change in the program parameters, and they are not erased, so they can be seen at the beginning of the text.

Lines 600-695 in the secondary program provide a listing of suggested beginning and ending line numbers for each page of the text you are composing, assuming an 11 inch paper length. You can copy these on a printer (if the COPY routine works for your printer, as it does with the 2040 printer). You may

want to adjust the values of "m" in lines 605 through 620 for your own preferences.

There are one or two cautions on using the system. If you should make extensive modifications to any text which had previously been recorded on a disc, the new text may not fit in the same disc space as the old one did. Then, assuming that you want to use the same name for the text, you should (1) go to BASIC, (2) erase the old text, then (3) return to Tasword and save the new version.

If you should accidentally try to print a text when the printer cable is not connected, you will lose control of the computer. Turning the computer on and then off will lose any text, so that's no help. The following procedure worked for me: turn the printer off, then carefully plug in the printer cable, and then turn the printer on. This should cause control to return to the keyboard.

Whenever you deliberately go to BASIC, or have some type of system error, you can return to Tasword by executing RUN.

The only "glitch" I know of in the program is that when saving or loading a file on disc, a "memory full" message is occasionally given. In that case, type RUN, and then try the operation again. That has always worked for me. One other curious thing occurred twice as I was doing some program development. After an operation which broke into the program, the cursor disappeared. After trying several other actions, I found that the repeated use of the left arrow eventually caused the return of the cursor. The Aerco System problem of sometimes failing to show a diskette index occasionally appears when using this program. If so, and you really need the index, then load in a fake file (such as xxx) through the normal Tasword procedures. This will give an error message. Then enter RUN and you should be back in normal operation. If you don't do this, normal operation will return later anyways.

This program works extremely well. It is a word processor capable of professional use. I'm quite familiar with the Wang word processing system (probably the best there is), and although this one doesn't have as many features, I find that I can accomplish nearly everything using this that I can with the Wang, and the speed of operation is nearly as great for most operations. The limit of 300 lines is a bit troublesome with Tasword, but is made less of a problem through the speed of the disc system.

FURTHER COMMENTS

It should be clear that the method of using two BASIC programs, loaded separately, is not the best possible way to use the Aerco system. It would be more elegant to make use of the additional dock memory to provide the extra needed memory. One

method would be to put one of the two BASIC programs into the dock, and then switch to that memory when that facility was selected. This would just provide a slight operational advantage, as it only takes about a second to read and load either of the BASIC programs from the disc. A second dock method would be to put the BASIC programs all in one of the memory banks, and the two digital files in the other bank. This would allow an expansion of the text file length beyond 300 lines. However, neither of those modifications has yet been developed, and the one shown here works now.

AN EXCELLENT PRINTER SYSTEM

Since it is so relevant to the discussion of the use of a word processor, I would like to call readers' attention to a printing system of which they probably are not aware. It is the Okidata Microline 82A, modified using the "PC Writer Printer Enhancement" made by Rainbow Technologies Inc. (17971 Skypark Circle, Suite E, Irvine, CA 92714). This enhancement was designed to work with an IBM PC, but it works extremely well with the Tasword II word processor. The various printer commands nicely fit the graphics command system of Tasword. Many commands have the same call sequence as in the Epson FX80, for which Tasword is originally set up; and other commands, up to a maximum of 16, can easily be written into Tasword. The letter quality printing mode is one of the very best available from a dot matrix printer: close to the quality of a daisy wheel printer. The speed of printing in letter quality mode is only a moderate 30 characters per second, and that mode is a bit noisy, but one can use the rapid 120 cps draft quality mode for all but the final copies. There also are many additional characters available, if you can figure out how to use those. Furthermore, dot addressable graphics is provided with the PC Writer. If you have an Okidata ML-82A, or ML-83A, or can get one of these inexpensively, I recommend the PC Writer, at $119.

© 1985, W. L. Fadner